NODE.JS NIGHTS

STRV

DATABASES

Samuel Prado, Backend Developer at STRV

STRV

AGENDA

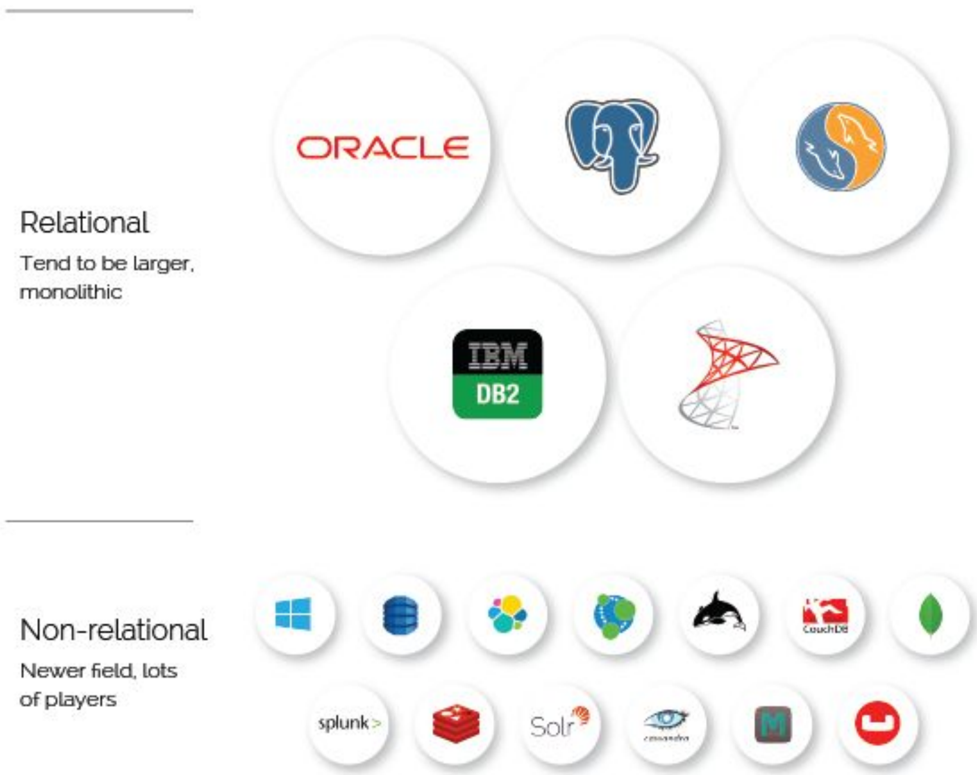
- Introduction
- Database types
- Install
  - Docker (for development)
- ORMs

INTRODUCTION

DATABASE TYPES

DATABASE TYPES

- Relational Databases
- Non-relational databases
    - NoSQL



Relational

Tend to be larger, monolithic

Non-relational

Newer field, lots of players

RELATIONAL DATABASES

RELATIONAL DATABASES

- Emerged in the 70’s
- Store data according to a schema
  - Data definition
- Allows data to be displayed as tables with rows and columns
- Provide functionality for reading, creating, updating, and deleting data
  - Data manipulation/query
  - **S**tructured **Q**uery **L**anguage

RELATIONAL DATABASES

Advantages

- Well-documented and mature technologies
- SQL standards are well-defined and commonly accepted
- Have ACID-compliant transactions
    - **A**tomicity
    - **C**onsistency
    - **I**solation
    - **D**urability

RELATIONAL DATABASES

Disadvantages

- Don’t work well — or at all — with unstructured or semi-structured data
  - Schema and type constraints
  - Non-suitable for large analytics or IoT event loads
- When migrating to another RDBMS, schemas and types must generally be identical between source and destination tables for migration to work

NON-RELATIONAL DATABASES

NON-RELATIONAL DATABASES

- Existed since the late 60’s
- Can be schema agnostic
  - Allow unstructured and semi-structured data to be stored and manipulated
- Increasingly used in **big data** and **real-time web** applications
- Have different types to solve different problems/needs

NON-RELATIONAL DATABASES

Types

- Key-Value Stores
  - Store only key-value pairs
  - Provides basic functionality for retrieving the value associated with a known key
- Wide Column Stores (Big Data)
  - Schema-agnostic
  - Works as a multi-dimensional key-value store

NON-RELATIONAL DATABASES

Types

- Document Stores
  - Schema-free
  - Store data in form of JSON documents
- Graph Databases
  - Represent data as a network of related nodes (or objects)
  - Typically used when analysis of the relationships between nodes is the end goal of the system

NON-RELATIONAL DATABASES

Advantages

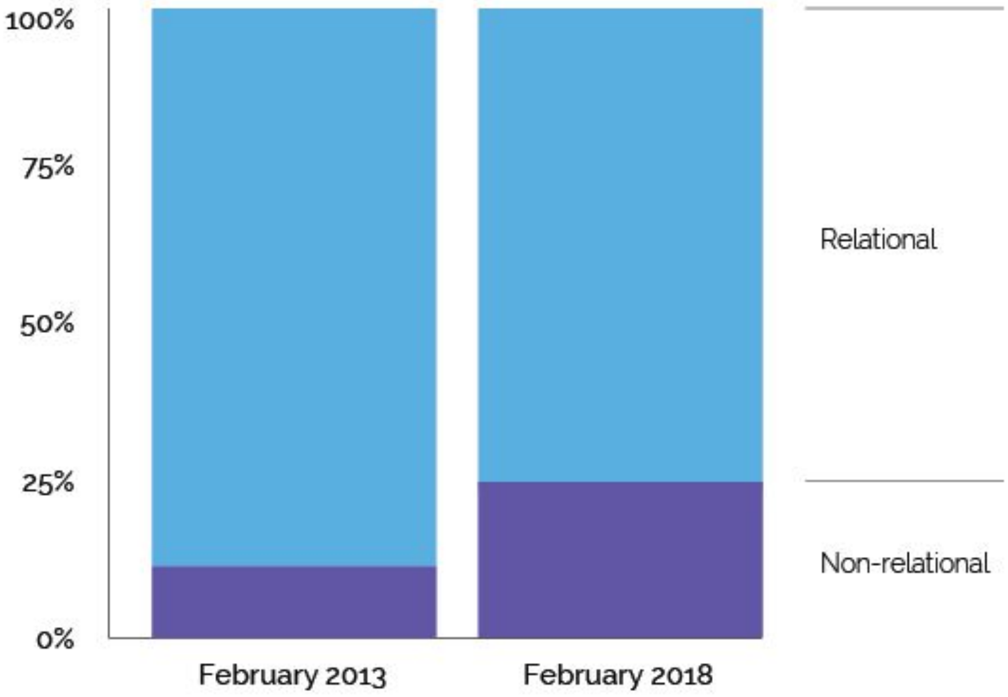
- Schema-free data models
  - More flexible and easier to administer
- Generally more horizontally scalable and fault-tolerant
- Data can easily be distributed across different nodes
  - To improve availability and/or partition tolerance, you can choose that data on some nodes be "eventually consistent"

NON-RELATIONAL DATABASES

Disadvantages

- Lack of relationships
- Data consistency needs to be taken care in the application logic
- Are generally less widely adopted and mature than RDBMS solutions
- Each database type have specific formats and constraints

POPULARITY - RDBMS vs. NoSQL



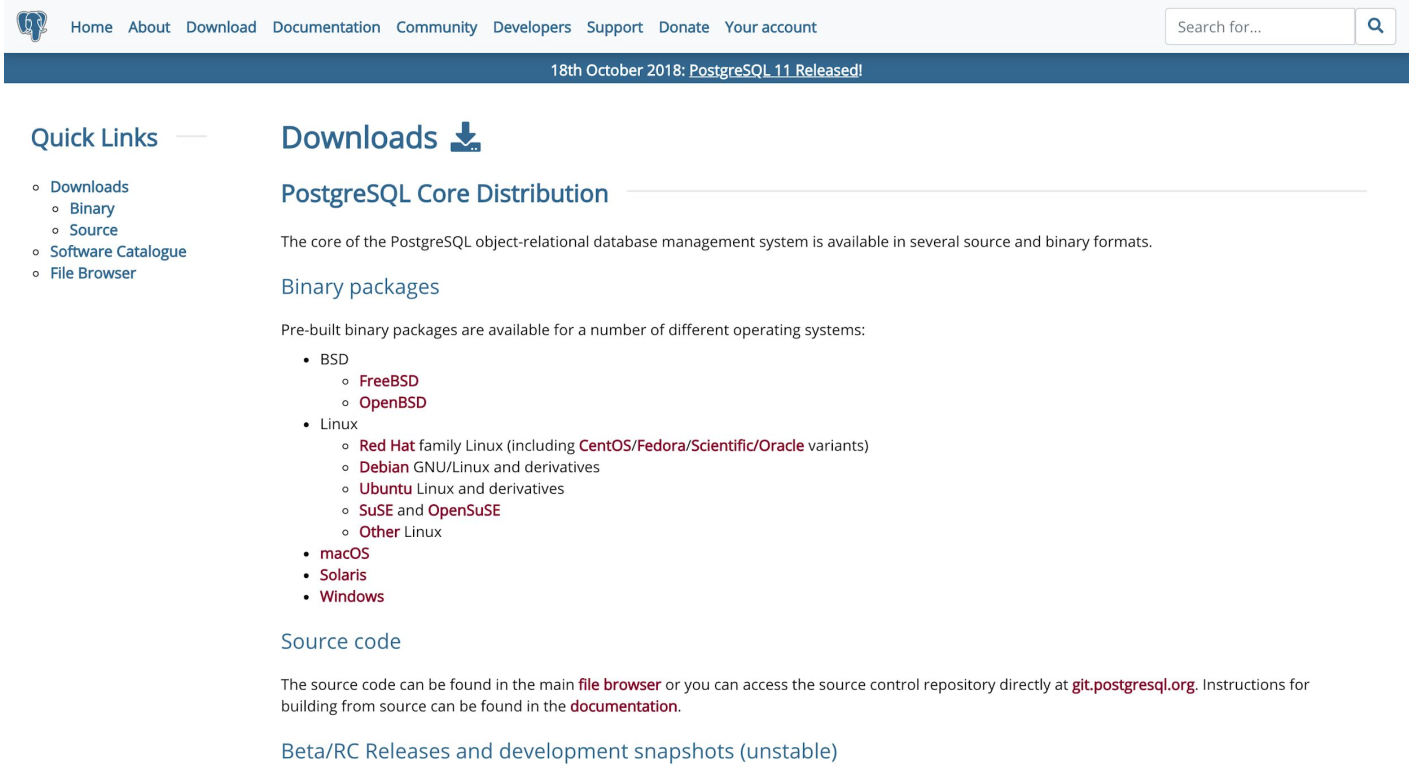
Source: https://db-engines.com/en/ranking_trend

INSTALL

INSTALL

- Which one???

PostgreSQL
the world's most advanced open source database



18th October 2018: PostgreSQL 11 Released!

Quick Links

- Downloads
  - Binary
  - Source
- Software Catalogue
- File Browser

Downloads

PostgreSQL Core Distribution

The core of the PostgreSQL object-relational database management system is available in several source and binary formats.

Binary packages

Pre-built binary packages are available for a number of different operating systems:

- BSD
  - FreeBSD
  - OpenBSD
- Linux
  - Red Hat family Linux (including CentOS/Fedora/Scientific/Oracle variants)
  - Debian GNU/Linux and derivatives
  - Ubuntu Linux and derivatives
  - SuSE and OpenSuSE
  - Other Linux
- macOS
- Solaris
- Windows

Source code

The source code can be found in the main file browser or you can access the source control repository directly at git.postgresql.org. Instructions for building from source can be found in the documentation.

Beta/RC Releases and development snapshots (unstable)

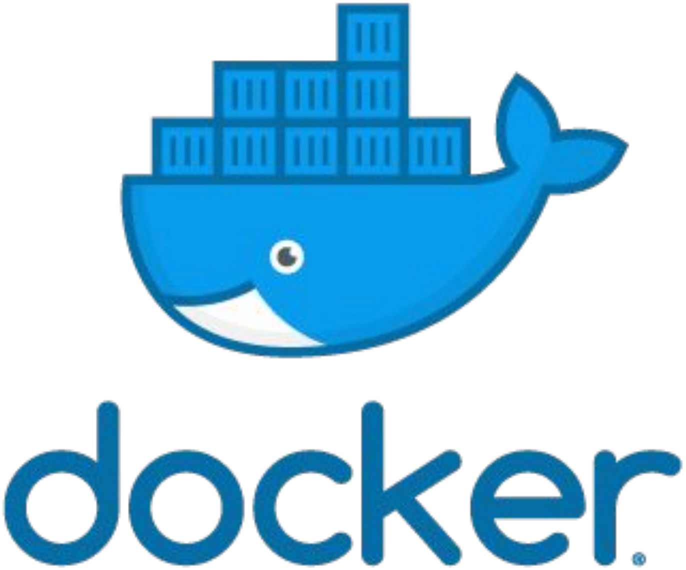
https://www.postgresql.org/download/

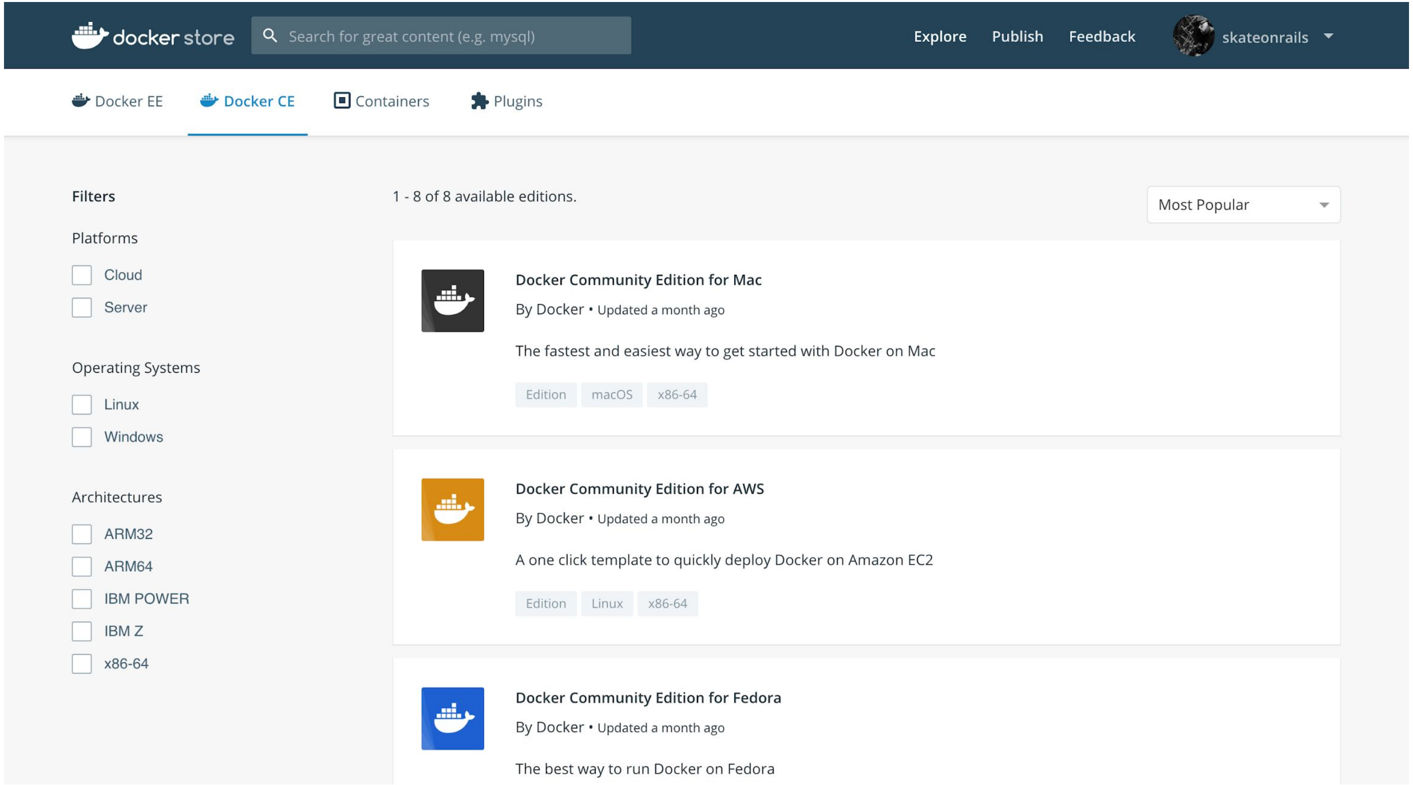
🎉 CONGRATULATIONS!!! 🎉

DOCKER

DOCKER

- Run software packages called "containers"
- Isolated from each other
- Bundle their own tools, libraries and configuration files
- More lightweight than virtual machines





Filters

Platforms

□ Cloud
□ Server

Operating Systems

□ Linux
□ Windows

Architectures

□ ARM32
□ ARM64
□ IBM POWER
□ IBM Z
□ x86-64

1 - 8 of 8 available editions.

Most Popular

Docker Community Edition for Mac
By Docker • Updated a month ago
The fastest and easiest way to get started with Docker on Mac
Edition · macOS · x86-64

Docker Community Edition for AWS
By Docker • Updated a month ago
A one click template to quickly deploy Docker on Amazon EC2
Edition · Linux · x86-64

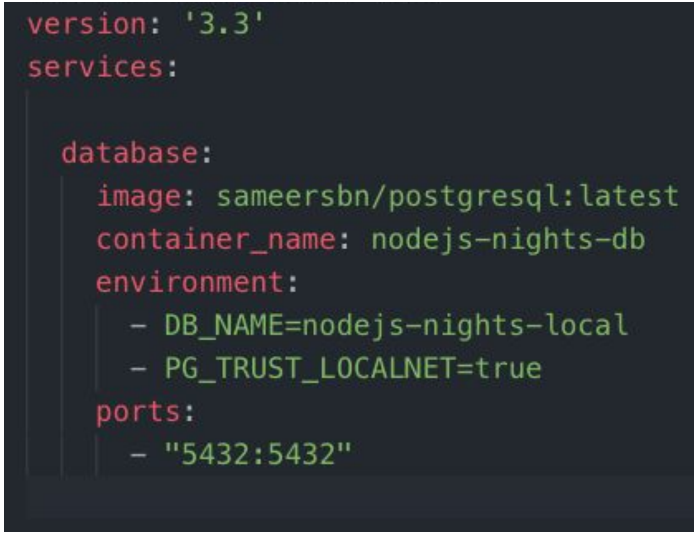
Docker Community Edition for Fedora
By Docker • Updated a month ago
The best way to run Docker on Fedora

https://store.docker.com/search?type=edition&offering=community

🎉 CONGRATULATIONS!!! 🎉

DOCKER-COMPOSE.YAML

- Create it on the project root folder
- Run
  - docker-compose up -d
  - You can set this command as an npm script



```yaml
version: '3.3'
services:

  database:
    image: sameersbn/postgresql:latest
    container_name: nodejs-nights-db
    environment:
      - DB_NAME=nodejs-nights-local
      - PG_TRUST_LOCALNET=true
    ports:
      - "5432:5432"
```

🎉 CONGRATULATIONS!!! 🎉

SETUP PROJECT

ORM

ORM

- **O**bject-**R**elational **M**apping
- Represents database rows (and it’s relationships) as objects
- Wraps implementation-specific details in a common API
- Helps to change the DBMS if needed
- Helps to maintain the database structure in a “timeline” (database migrations)

ORM

- Sequelize
- Bookshelf
- Objection.js

SETUP PROJECT (AGAIN)

HOMEWORK

- Set a database to your project
- Update the repository layer with database CRUD operations
- Use an ORM of your choice

THAT'S IT

Samuel Prado
samuel.prado@strv.com

STRV

QUESTIONS

STRV